

**REMARKS/ARGUMENTS**

Claims 1-26 were pending in the present application when last examined. No claims have been added or amended and no new matter has been added. Therefore, upon entry of this amendment, which is respectfully requested, claims 1- 26 will be pending.

**Claim Rejections under 35 USC § 102(e) over Lin**

On page 2 of the Office Action, claims 1-4, 20, 22 and 24-26 have been rejected under 35 U.S.C. 102(e) as being anticipated by Lin, U.S. Patent Application Publication No. US 20050071345 (hereinafter "Lin"). Applicants respectfully traverse and request withdrawal of this rejection for at least the following reasons.

Regarding claim 1, Applicant respectfully submits that the rejection fails to properly consider ALL limitations of claim 1, the proper consideration of which would yield a contrary result, and is therefore improper. Moreover, Lin not only fails to anticipate, but further teaches away from the embodiment recited by claim 1.

Present Claim 1 recites:

“1. (Previously Amended) A computer-implemented method of storing multiple fields for multiple tenants in a single multi-tenant data structure, comprising:  
defining a multi-tenant data structure having a plurality of data columns and one or more index columns;

defining a first data field for a first tenant, said first field having a first data type;

defining a second data field for a second tenant, said second field having a second data type, wherein the second data type may be different than said first data type; and

when records having data values in the first and second fields are created by the first and second tenants, storing the data values of first and second fields to a single column in the data structure, wherein the single column includes data values that may include different data types for different tenants” (emphasis added).

The Examiner asserts that Lin discloses “a data structure having a plurality of data columns and one or more index columns,” i.e., a relational database. However, the Examiner fails to consider limitations of present claim 1 including the ALSO recited “... method of storing multiple fields for multiple tenants in a single multi-tenant data structure” and the ALSO-recited defining of “a multi-tenant data structure having a plurality of data columns and one or more index columns” (emphasis added). Having ignored such limitations, the Examiner also fails to properly consider: (1) *whether* Lin anticipates a multi-tenant data structure or related claim 1 limitations, and (2) that Lin *clearly* fails to do so.

Exemplary multi-tenant data structures are discussed in the instant specification beginning with the para. 0002, which provides: “In multi-tenant database systems, such as the salesforce.com service, a multi-tenant architecture is used wherein customer organizations (i.e., tenants) share database resources in one logical database” (emphasis added). nowhere in the cited Lin paragraphs 28 and 31 or elsewhere does Lin teach, suggest, otherwise anticipate or even mention one or more tenants. Nowhere does Lin consider let alone anticipate that a disclosed data structure may be defined or used in conjunction with more than one organization (“i.e., tenant”), let alone particularities corresponding thereto. Rather, Lin merely discloses techniques that “allow users to store data for... custom attributes of application objects in repository tables *without adding new columns to the existing tables,*” “*to upgrade the application*” and to “[retrieve] data for custom object types and... custom attributes” (para 0026, emphasis added). Not only is the entirety of Lin directed instead at **only** a single group of one or more closely related users using the same application for the same purposes (see, for example, Lin FIGS. 1 through 4), but attempts to apply the teachings of Lin to multiple tenants or a multi-tenant data structure of claim 1 may well thwart the goals of Lin, the claim 1 embodiment or both (as is discussed next).

For example, assuming *arguendo* that the storing, customizing, upgrading and/or retrieving techniques according to Lin are applied to a multi-tenant data structure, the same storing, customization or upgrade of **any one** tenant organization will necessarily and undesirably impose the very **same** and potentially incompatible customization or upgrade on **all other** tenant organizations. Data retrieval by **one tenant** would further necessarily and undesirably retrieve potentially proprietary information of **all other** organizations or “tenants,” and so on. Lin simply does **not** consider, is not equipped and may **well** thwart the recited multiple tenants and multi-tenant data structure limitations of claim 1 by preventing practical utilization by the recited multiple tenants.

The techniques of Lin also provide that “the number of custom-attribute tables... increases relative to the number of data types of the custom attributes” (para. 0029). Thus, if we assume *arguendo* that the recited limitations of claim 1 are imposed on Lin’s teachings (that instead relate to facilitating single user group customization of objects of a shared application), then the number of tables will likely be multiplied by at least **each** of: **multiple** tenant organizations, **multiple** applications that are more likely used by such tenant organizations, **multiple** groups within the multiple tenant organizations, and **multiple** users within the multiple groups of each of the multiple tenant organizations. As a result, requisite data structure size, needed resources and complexity may quickly render the data structure unsuitable according to **both** Lin and the claim 1 embodiment, among other potential difficulties. Moreover, Lin simply does **not** mention, let alone anticipate any aspect of such imposition, which must therefore arise from impermissible hindsight.

Worse yet, having neglected to consider the above limitations also apparently led the Examiner to also neglect or to misconstrue further claim 1 limitations. For example, the Examiner asserts that Lin discloses “defining a first data field for a first tenant,”... (See paragraph 0028), “defining a second data field for a second tenant,”... (See paragraph 0028)” and “when records having data values in the first and second fields are created by the first and second tenants, storing the data

values of first and second fields to a single column in the data structure... (see paragraph 0031) (emphasis added).

However, as discussed above, Lin clearly does **not** mention or consider, let alone anticipate a multi-tenant data structure. Lin also does not mention or consider, let alone anticipate multiple tenants, “defining a... data field for a first tenant,” “a second tenant” **or** multiple tenants (e.g., first and second tenants) in either the cited paragraph 0028 or elsewhere in Lin. Applicant presumes, from the Examiner’s assertion, that the Examiner either failed to consider or incorrectly associated the recited term “tenant” with “user.” However, a “tenant” organization, customer organization, well-known hosted customer organization provided by salesforce.com (see, for example, present para. 0002) or other similar service, and so on, are clearly **not** the individuals of Lin (see also all Lin figures). Moreover, Lin’s teachings as to individuals modifying a database used by only the individuals for only a common application manipulating only common data for only common purposes clearly does **not** relate to, let alone anticipate: defining a multi-tenant data structure within which storage locations may be defined for different organizations, data stored for different organizations, multiple tenant data/different data types integrated in a single column (see below), and so on that may relate to multiple unrelated organizations that may use different applications to store/manipulate different data for different purposes.

The Examiner also asserts that Lin teaches “when records having data values in the first and second fields are created by the first and second tenants, storing the data values of first and second fields to a single column in the data structure,” “*wherein the single column includes data values that may include different data types*” and that the single column includes data values that may include different data types “*for different tenants*” (See paragraph 0031)” (emphasis added). Applicant respectfully disagrees.

First, Lin not only fails to teach or suggest a “single column includes data values that may include *different data types*,” (let alone different data types “for different tenants”). Rather, Lin directly contradicts the Examiner’s assertion; Lin

instead teaches that different data types are not merely stored in separate *columns*, but “data for custom attributes of each data type is stored in separate tables” (para 0061), such that “a single column is provided for storing data for all custom attributes of the same data type” (Cited paragraph 0028), and “a separate custom-attribute table is established for storing the values of each of the different data types (Lin para. 0029) (emphasis added). Thus, the number of custom-attribute tables... increases relative to the number of data types of the custom attributes,” for example, requiring “nine distinct custom attribute tables” where each of “nine custom attributes has a different data type” (Lin para. 0029).

Cited Lin para. 0031 also fails to support the Examiner’s assertion. Para. 0031 discusses that “a custom-attribute table includes a number of columns” that “in one embodiment... includes one or more (each of) instance identifying... attribute-identifying... and... value columns.” However, Lin para. 0031 is silent as to the particular content of each column, and the proper result must therefore be read in accordance with the preceding Lin para. 0028-0029, Lin para. 0030 (“data for a second custom attribute of the same object type may be stored in a second table based on the fact that the second custom attribute has a second data type that is different from the first data type”) and the following Lin para. 0032 (“As mentioned above, each custom-attribute table is associated with a data type”). Cited Lin para. 0031 further fails to support the Examiner’s assertion that a single column in Lin includes data values that may include different data types for or is in any other way related to “different tenants,” since it is completely devoid of **any** teaching respecting different tenants.

It is therefore respectfully submitted that claim 1 is patentable over Lin for at least the foregoing reasons. Claims 2 through 4 and 24 through 26 further depend from claim 1 and are patentable over Lin for at least the same reasons that claim 1 is patentable over Lin. The Examiner further asserts that claims 20, 22 comprise substantially the same limitations as claim 1, and is thus rejected for the same reasons as set forth in the rejection of claim 1. Applicant therefore respectfully submits that

claims 20, 22 are patentable over Lin for at least the same reasons that claim 1 is patentable over Lin. Withdrawal of the rejections of claims 1, 2-4, 20, 22 and 24-26 and early allowance of the same is solicited.

### **Claim Rejections under 35 USC § 102(e) over Millet**

On page 5 of the Office Action, claims 5-19, 21 and 23 have been rejected under 35 U.S.C. 102(e) as being anticipated by Millet, U.S. Patent Application Publication No. US 20030154197 (hereinafter "Millet"). Applicants respectfully traverse and request withdrawal of this rejection for at least the following reasons.

Regarding claim 5, Applicant submits that the rejection fails to properly consider **all** limitations of claim 5, the proper consideration of which would yield a contrary result, and is therefore improper. Moreover, Millet not only fails to anticipate, but further teaches away from the embodiment recited by claim 5.

Present Claim 5 recites:

5. (Original) A computer-implemented method of hosting multiple tables for one or more organizations in a single multi-tenant data structure, comprising:

defining a multi-tenant data structure having a primary key column, an organization id column and a plurality of data columns;

defining a first table for a first tenant, said first table having a first data field, and said first tenant having a first tenant id;

assigning a first table id to the first table;

defining a second table for a second tenant, said second table having a second data field, and said second tenant having a second tenant id;

assigning a second table id to the second table;

wherein when records are created for the first table by the first tenant, for each created record:

a) storing the value of the first data field to a single data column in the data structure;

b) storing the first tenant id in the organization id column; and

c) storing the first table id to the primary key column; and

wherein when records are created for the second table by the second tenant, for each created record:

a) storing the value of the second data field to said single data column in the data structure;

b) storing the second tenant id in the organization id column; and  
c) storing the second table id to the primary key column; and  
wherein the first and second tables of the first and second tenants are  
stored in the data structure. (emphasis added)

As with the above rejections over Lin, the Examiner again fails to consider **all** claim 5 limitations. For example, the Examiner fails to consider that claim 5 is directed to “a computer implemented method of hosting multiple tables for one or more organizations in a single multi-tenant data structure,” which are **not** taught or suggested in **any** of the cited Millet paragraphs or elsewhere in Millet. The Examiner asserts that Millet discloses “defining a multi-tenant data structure having a primary key column (See paragraph 0042), an organization id column and a plurality of data columns.” However, such argument: completely fails to address the “hosting” limitation, which Millet fails to anticipate; does not adequately address the remaining ignored limitations; and belies that Millet simply fails to teach or suggest and further teaches away from the recited limitations.

For example, Millet is directed at “storing a set of data records traditionally represented in a flat RDBMS data table in a linked series of tables” to “permit the user to change the structure of the data records without requiring modifications to the application software or the associated data table structures implemented in the RDBMS” (cited Millet para. 0042) (emphasis added). Such traditionally represented data records, which are illustrated beginning with Millet FIG. 1 and the cited para. 0042, clearly show a traditional home or business relational database storing traditional data relating to (and further simply identifying) employees of a single company. Such teachings clearly do **not** teach or suggest and contrast sharply with at least the recited “... hosting multiple tables for one or more organizations in a single multi-tenant data structure” or “defining a multi-tenant data structure having a primary key column, an organization id column and a plurality of data columns,” which were not traditionally known. The cited and remaining Millet teachings also disclose using a traditional “Employee ID” as a “unique identifier for a particular data record” and as a primary key for querying employee records of company employees in a traditional single company database (Millet para. 0042, 0042 and thereafter, related Millet FIGS. 1-4). These teachings also clearly do **not** teach or suggest

and are in sharp contrast to the recited “organization ID” that may be stored in an “organization column,” for example, to indicate various data records relating to a “first tenant” and a “second tenant” in a “multi-tenant data structure.”

Moreover, assuming arguendo that the recited limitations might somehow be imposed on Millet, Millet would nevertheless teach away from the recited embodiment and may well render both Millet and the claim 5 embodiment inoperable. For example, Millet more specifically explains that: “the flexibility of the database application is increased by disaggregating the information usually stored within a single database table” into “a series of three or more database tables” (Millet para. 0048). One must therefore assume that, if multiple tenant data (or a multiple tenant data structure) is imposed on Millet, then for consistency, Millet would further similarly disaggregate the multiple tenant data. First, Claim 5 conversely provides for integrating multiple tenant data within a single column in a multiple tenant data structure (“storing the value of the first data field” corresponding to the first tenant and “storing the value of the second data field” to the same “single data column in the data structure.”) Further, such disaggregating of multiple tenant data would prevent such integration and may render benefits of the embodiment of claim 5 arising therefrom inoperable. Moreover, such disaggregating of multiple tenant data that may correspond to **each** of: **multiple** tenant organizations, **multiple** applications, **multiple** groups within the multiple tenant organizations, and **multiple** users within the multiple groups of each of the multiple tenant organizations may well render data structure size, needed resources and complexity impracticable for **both** Millet and the present claim 5 embodiment.

The Examiner also asserts that Millet discloses “defining a first table for a first tenant, said first table having a first data field, and said first tenant having a first tenant id; assigning a first table id to the first table (See paragraph 0054)” and “defining a second table for a second tenant, said second table having a second data field, and said second tenant having a second tenant id; assigning a second table id to the second table” (emphasis added). Applicant respectfully disagrees.

As with the above claim 1 rejection over Lin, having neglected to consider **all** limitations apparently led the Examiner to also neglect or to misconstrue further claim 5



limitations. As discussed, Millet clearly does **not** mention or consider, let alone anticipate a multi-tenant data structure. Millet further does **not**, in the cited Millet para. 0054 or elsewhere, anticipate a “first tenant” and a “second tenant,” let alone “defining a first table for a first tenant” and a second table “for a second tenant,” or further, where “multiple tables for one or more organizations in a single multi-tenant data structure” are hosted. Millet also does **not** anticipate a “first tenant ID” or a “second tenant ID,” let alone first and second tenants having first and second tenant IDs respectively. Moreover, cited Millet para. 0042 teaches a one to one correspondence between an Employee ID and that employee’s one matching data record (and **not** a recited tenant or “organization” ID that may correspond with one or more different data records, or further, one or more records corresponding to a particular tenant) Cited paras. 0054-0055, in considering IDs, **only** discuss a “row ID” and a “Table ID,” but do **not** discuss the asserted first or second tenant or first or second tenant ID of claim 5. (While Millet para. 0054 also discusses primary and foreign keys, these relate to the traditional database of Millet FIG. 2, are unrelated and provide **no** support for the Examiner’s assertion.)

Applicant presumes, from the Examiner’s assertion, that the Examiner again either failed to consider or incorrectly associated the recited term “tenant” with some other inconsistent term or terms in Millet. However, a “tenant” organization, customer organization, hosted customer organization provided by salesforce.com or other similar service (see, for example, present para. 0002), and so on, are clearly **not** used, taught or suggested by Millet.

The Examiner also asserts that Millet discloses “wherein when records are created for the first table by the first tenant, for each created record (See paragraph 0054): a) storing the value of the first data field to a single data column in the data structure; b) storing the first tenant id in the organization id column; and c) storing the first table id to the primary key column; and wherein when records are created for the second table by the second tenant, for each created record: (See paragraph 0055) a) storing the value of the second data field to said single data column in the data structure; b) storing the second tenant id in the organization id column; and c) storing the second table id to the primary key column; and wherein the first and second tables of the first and second tenants are stored in the data structure (See paragraph 0055). Applicant respectfully disagrees.

As was already discussed: Millet **clearly** fails to anticipate a “first tenant” and a “second tenant,” let alone a “first tenant ID” and a “second tenant ID,” an “organization ID column” or **any** other correspondence between respective tenants and IDs; Millet also **clearly** fails to anticipate storing values corresponding to a “first tenant” and “second tenant” and teaches disaggregating into *different tables* data that was traditionally stored in one database table (Millet para. 0048) rather than the contradictory recited storing of such values in a *same column*, let alone the totality of the recited limitations of claim 5. Therefore, Millet not only fails to anticipate, but further teaches **away** from the limitations of claim 5. Applicant also reiterates that Millet provides **no** teaching or suggestion in the cited paragraphs 0042, 0054-0055 or elsewhere that even mention, let alone anticipate the recited limitations.

It is therefore respectfully submitted that claim 5 is patentable over Millet for at least the foregoing reasons. Claims 6 through 8 depend from claim 5 and are patentable over Millet for at least the same reasons that claim 5 is patentable over Millet. Withdrawal of the rejections and early allowance of claims 5-8 is respectfully requested.

According to the Examiner, “claim 9 comprises substantially the same limitations as claim 5 and is thus rejected for the same reasons as set forth in the rejection of claim 5.” Therefore, assuming *arguendo* that the Examiner is correct, it is respectfully submitted that claim 9 is patentable over Millet for at least the same reasons that claim 5 is patentable over Millet. Claims 10 through 19 further depend from claim 9 and are patentable over Millet for at least the same reasons that claim 9 is patentable over Millet. Withdrawal of the rejections and early allowance of claims 9-19 is respectfully requested.

According to the Examiner, “claims 21 and 23 comprise substantially the same limitations as claim 5 and is thus rejected for the same reasons as set forth in the rejection of claim 5.” Therefore, assuming *arguendo* that the Examiner is correct, it is respectfully submitted that claims 21 and 23 are patentable over Millet for at least the same reasons that claim 5 is patentable over Millet. Withdrawal of the rejections and early allowance of claims 21 and 23 is therefore respectfully requested.

Appl. No. 10/817,161  
Amdt. dated November 26, 2007  
Reply to Office Action of July 27, 2007

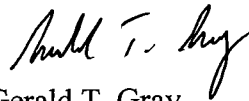
PATENT

**CONCLUSION**

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance and an action to that end is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 925-472-5000.

Respectfully submitted,



Gerald T. Gray  
Reg. No. 41,797  
on behalf of Daryl Josephson  
Reg. No. 37,365

TOWNSEND and TOWNSEND and CREW LLP  
Two Embarcadero Center, Eighth Floor  
San Francisco, California 94111-3834  
Tel: 925-472-5000  
Fax: 415-576-0300  
Attachments  
GTG:sea  
61206606 v1